Towards 3D Image Reconstruction: 3D Visualization from 2D Ultrasound Images

Mini Project Report

Submitted by

Dan Mathews Robin

Parvathy H

Sachin V S

Swathy Prasad

In partial fulfillment of the requirements for the award

of Master of Science in Computer Science of

Cochin University of Science and Technology, Kochi

under the guidance of

Dr. Jose Joseph

conducted by

Department of Machine Intelligence

Indian Institute of Information Technology and Management-Kerala

Technopark Campus

Thiruvanathapuram-695 581

January 2022

DECLARATION

We undersigned hereby declare that the project report "Towards 3D Image Reconstruction: 3D Visualization from 2D Ultrasound Images" , submitted for partial fulfillment of the requirements for the award of degree of Master of Science of the Cochin University of Science and Technology, Kerala is a bonafide work done by us under supervision of Dr. Jose Joseph. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources to our best. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Dan Mathews Robin
Parvathy H
Sachin V S
Swathy Prasad

Place: Thiruvananthapuram
Date: 05/01/2022

Department of Machine Intelligence

Indian Institute of Information Technology and Management-Kerala

Thiruvanathapuram

2020 - 2022

CERTIFICATE

This is to certify that the report entitled **Towards 3D Image Reconstruction: 3D Visualization from 2D Ultrasound Images** submitted by **Dan Mathews Robin, Parvathy H, Sachin V S, Swathy Prasad**, to the Cochin University of Science and Technology in partial fulfillment of the Master of Science degree in Computer Science is a bonafide record of the project work carried out by them under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mini Project Coordinator

Dr. Jose Joseph
Assistant Professor
IIITM-K

Head of the Department

Prof. Asharaf S
Professor
Dept. of Machine Intelligence
IIITM-K

ACKNOWLEDGEMENT

We take this opportunity to express our deep sense of gratitude to all who helped us to complete the work successfully. Our first and foremost thanks goes to God Almighty who showered in immense blessings on our effort.

We wish to express our sincere thanks to **Dr. Jose Joseph** for providing us with all the necessary facilities and support and valuable inputs.

We would also like to thank **Francis Kalloor Joseph PhD** (postdoctoral fellow at Biomedical Photonic Imaging group in the Faculty of Science and Technology, University of Twente) for his valuable inputs.

We would like to express our sincere gratitude to **Prof. Asharaf S(Head of the department)**, for his support and co-operation. We wish to express our sincere gratitude towards all the teaching and non teaching staff members of our Department.

Finally we thank our parents, all our friends, near and dear ones who directly and indirectly contribute to the successful of this work.

Dan Mathews Robin

Parvathy H

Sachin V S

Swathy Prasad

Abstract

Three dimensional (3D) ultrasound image reconstruction from two dimensional (2D) images is a suitable method for analyzing some anatomy related abnormalities.Ultrasound image reconstruction system is required in order to view the specific part of an object so that it can be used for analysis purpose .In this project we have taken few sequences of ultrasound images of a conical frustum, a tube and a sponge as input. Few image processing techniques like thresholding, canny edge detection and K Means clustering were applied . After processing the images their 3D visualization was done by converting to point cloud file and surface is rendered using pyvista. In another approach, isosurface using marching cubes from VTK is used for 3D visualization . From our experiments we came to a conclusion that better filtering, region of interest segmenting techniques and more images are required for improved results.

Contents

List of Figures

ABBREVIATIONS

US Ultrasound

MRI Magnetic resonance imaging

CT Computed tomography

PCD Point Cloud Data

Chapter 1

Introduction

Three dimensional (3D) ultrasound image reconstruction from two dimensional (2D) images is a suitable method for analyzing some anatomy related abnormalities.Ultrasound image reconstruction system is required in order to view the specific part of an object so that it can be used for analysis purposes. Cost of a 3D Ultrasound machine is high and un affordable of lower income hospitals. Here we are exploring methods to convert 2D US images to 3D using basic image processing techniques, K mean clustering and marching cubes algorithm.

1.1 Background of 3D object reconstruction from 2D Images

3D reconstruction can be summarized as the process of inferring a shape that is as close as possible to the unknown shape. The systems can be categorized based on the nature of the input, the representation of the output etc. In particular, the input can be a single image, multiple images captured using devices whose intrinsic and extrinsic parameters can be known or unknown, or a video stream, i.e., a sequence of images with temporal correlation.

Some common output representations are:

- **Volumetric representations**, which have been extensively adopted in early deep leaning-based 3D reconstruction techniques, allow the parametrization of 3D shapes using regular voxel grids. As such, 2D convolutions used in image analysis can be easily extended to 3D. They are, however, very expensive in terms of memory requirements, and only a few techniques can achieve sub-voxel accuracy.

- **Surface-based representations**, such as meshes and point clouds have also been explored. While being memory-efficient, such representations are not regular structures and thus, they do not easily fit into deep learning architectures.

- **Intermediation,** in which some 3D reconstruction algorithms predict the 3D geometry of an object from images directly, others decompose the problem into sequential steps, each step predicts an intermediate representation.

Chapter 2

Literature Survey

In Raphael Prevost et al [5], they aimed at creating 3D freehand ultrasound reconstructions from 2D probes with image-based tracking, therefore eliminating expensive or cumbersome external tracking hardware. A convolutional neural network (CNN) design is brought into picture to directly estimate the motion of successive ultrasound frames in an end-to-end fashion.

In the paper by Knauer et. al [1] references to spatial compounding of time-lapse optoacoustic data and exploits the frequency domain properties of vascular networks in optoacoustic images and estimates the relative motion and orientation of the imaging probe. This allows rapidly combining sequential volumetric frames into large area scans without additional tracking hardware.[4] covers a systematic review of 3-D US imaging between 1970 and 2017, highlighting the current trends in research fields, the research methods, the main limitations, the leading researchers, standard assessment criteria and clinical applications.

As we began to study on images and their manipulation for our goal we came across few algorithms that proved quite useful in conducting our experiments. In [6] they describe canny edge algorithm.We have used this in identifying the edges of ultrasound images. In [3] they discuss about K means clustering algorithm. For this project we have used this algorithm for clustering image in order to remove its noise. In [2] they discuss on using binary thresholding for image segmentation. In our project we have done the same.The papers and review served as a starting point for us to get an rough overview of the problem that we were to address.

Chapter 3

Ultrasound Image Data

We received three sets of ultrasound images. The images were of a conical frustum, tube and a sponge. All the images were of bitmap (.bmp) format.

All images had been captured 2mm apart. Imaging had been done as described by the following diagram.

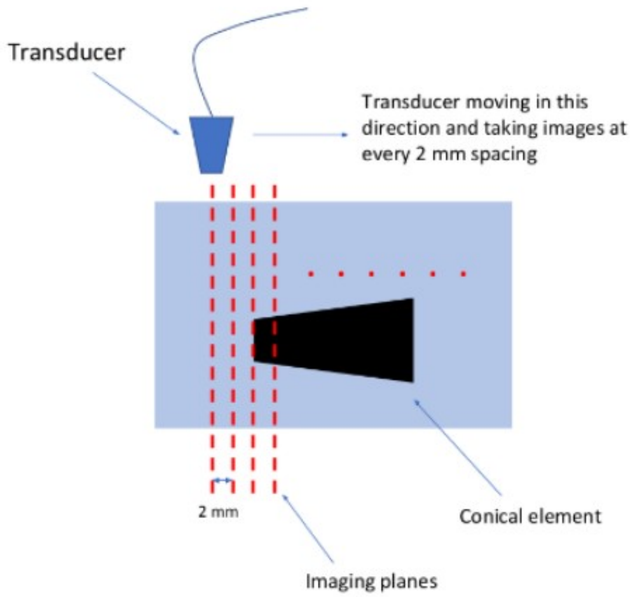

Figure 3.1: Imaging method

.

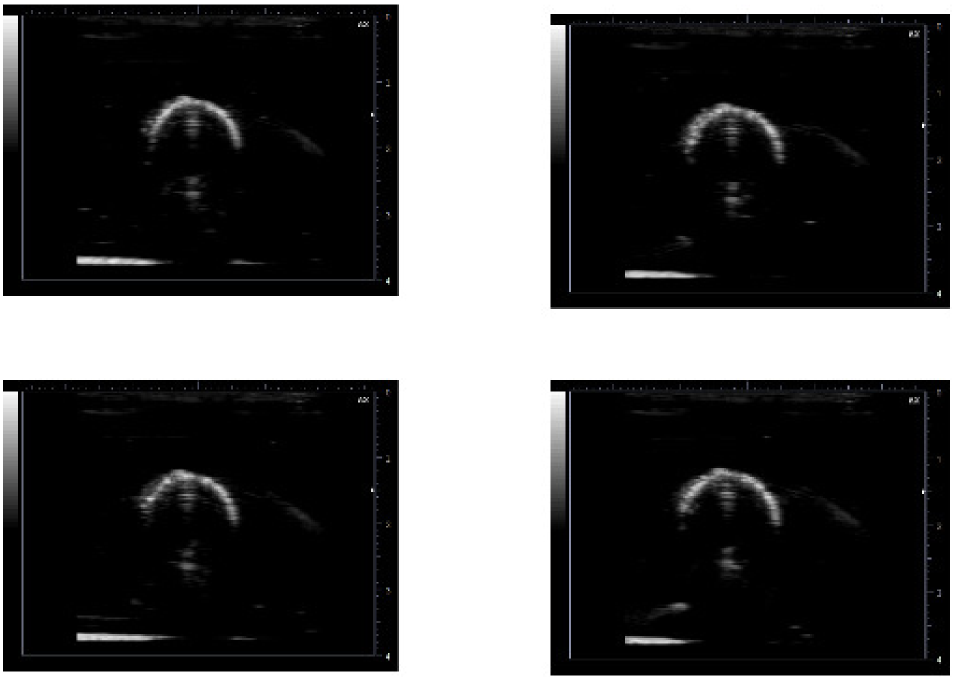

Figure 3.2: Some ultrasound slices of conical frustum

.

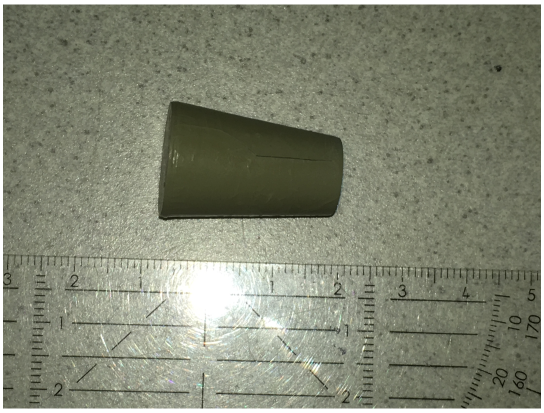

Figure 3.3: Conical frustum

.

As seen from the images, few of the data especially on its bottom side didn't get captured in ultrasound.This has affected the quality of final 3D image.

Chapter 4

Preprocessing

4.1 Image segmentation

Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analysing the image becomes simpler. We use various image segmentation algorithms to split and group a certain set of pixels together from the image. Here we have used few methods for segmentation.

4.1.1 Thresholding

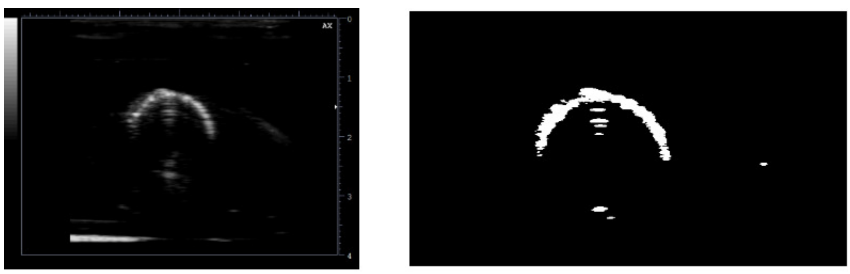

Figure 4.1: Binary thresholding

.

Thresholding will assign a predefined value , say white, to all pixels having intensity above threshold limit and a different value, say black, to all those pixels with intensities less than or equal to the threshold limit. This is simple binary thresholding.

4.1.2 K Means clustering algorithm

K means clustering is an unsupervised algorithm used to identify clusters in the data. K means clustering can be used in image data to segment interesting areas from the background. It clusters given data into K clusters using the k centroids.This algorithm is used when we have unlabelled data. The goal is to find certain groups based on some kind of similarity in the data with the number of groups represented by K. In case of K means

---

Algorithm 1: Algorithm for K means clustering

Input: Ultrasound images

Output: Clustered images

1 Choose number of clusters
2 Select random K points, K stands for number of centroids
3 Each data points will be assigned to closest centroids
4 Calculate and position each cluster's new centroid.
5 Each data point should be reassigned to the new nearest centroid. proceed to step 4, otherwise the model is complete.

---

finding the correct value of K is necessary. The Elbow method is used to find the correct value of K. The main principle underlying partitioning methods like K-Means clustering is to form clusters with the least amount of intra-cluster variation, or total within-cluster sum of squares (WCSS). We want the total WCSS to be as little as feasible because it measures the compactness of the grouping. Input of our project is a sequence of ultrasound images.It contains a lot of noise and our goal is to filter those noises using K means clustering. The K means algorithm is imported using the OpenCV library. After performing K Means clustering, it is visible that the image got segmented and noises which were present got removed.

Figure 4.2: Clustering with k=3

.

Figure 4.3: Clustering with k=2

.

4.1.3 Canny edge detection

Algorithm 2: Algorithm for Canny edge detection

Input: Clustered images

Output: Processed images

1 Gradient calculation

2 Non-maximum suppression

3 Double threshold

4 Edge Tracking by Hysteresis

OpenCV provides cv2.Canny(image, threshold1,threshold2) function for edge detection.The first argument is our input image. Second and third arguments are our min and max threshold respectively. Using the Canny algorithm, the function discovers edges in the input image (8-bit input picture) and marks them in the output map edges. For edge linking, the least value between threshold1 and threshold2 is chosen. The biggest value is used to locate the beginnings of strong edge segments.

Figure 4.4: Canny edge detection

.

Chapter 5

3D Visualization

5.1 Using VTK

VTK is an open source software package for computer graphics, available in python. VTK provides sample steps to convert CT DICOM image slices to 3D. This approach uses VTK's vtkImageData() method to arrange the DICOM slices as volume, and vtkMarchingCubes() to render the volume in 3D. vtkMarchingCubes() is an implementation of Marching cubes algorithm which helps to generate isosurfaces from volume [7] .

These steps are convenient to be used on 2D ultrasound image slices captured using a linear transducer probe. The ultrasound images we received were not DICOM, but were in the bmp format. To leverage the existing methods from VTK, we decided to convert the bmp images to DICOM (.dcm) format by adding dummy DICOM header to the bmp images. Main steps followed are shown below.

Algorithm

The process of 3D visualization of ultrasound bmp images using VTK has the following steps:

Algorithm 3: Algorithm to covert US bmp images to DICOM and to visualize in 3D using VTK

Input: Ultrasound images in bmp format

Output: 3D visualization of ultrasound images

```
Extract just the header from sample CT dicom image
Preprocess the US bmp images given
Set common fields of the dicom header with appropriate values
For each preprocessed US bmp image
Create a dicom file
Read the US bmp image pixel data
Provide the read in data as PixelData value for current dicom file
Update InstanceNumber field
Update ImagePositionPatient field
Save this current dicom data as .dcm file
Give the US dicom images obtained from previous step as input directory to
  vtkDICOMImageReader()
Generate volume of these dicom images using vtkImageData()
Generate isosurface from this volume for 3D visualization using
  vtkMarchingCubes()
```

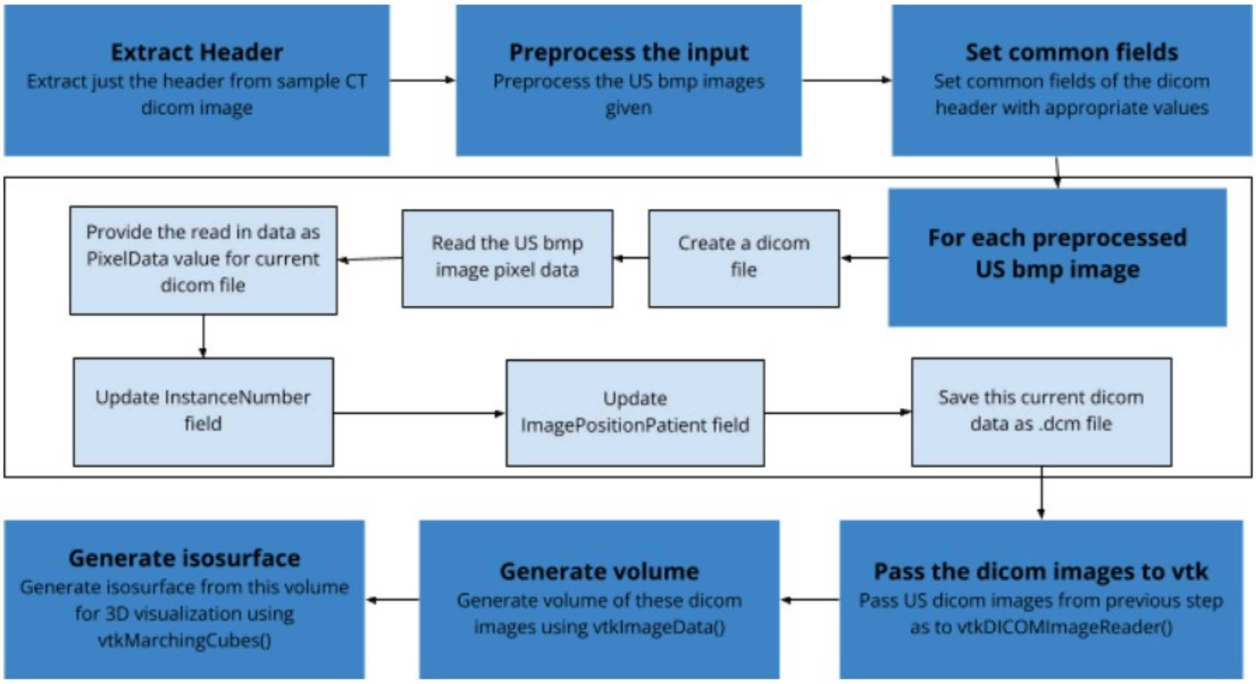

Figure 5.1: 3D visualization using VTK

5.2 Point cloud representation

A point cloud is a collection of data points that are arranged in space. A 3D shape or item could be represented by the points. Cartesian coordinates are assigned to each point position (X, Y, Z). 3D scanners or photogrammetry software, which measure multiple points on the external surfaces of things surrounding them, produce point clouds. Point clouds are created as a result of 3D scanning procedures and are used for a variety of applications, including creating 3D CAD models for manufactured parts, metrology and quality inspection, and a variety of visualisation, animation, rendering, and mass customisation. In our application we have a stack of ultrasound image sequences of an object and since we have segmented and identified its edges we can convert it into a 3 D point cloud file. With gap between them representing the gap between ultrasound slices.To convert our data into point cloud, we use a library called Open3D. Open3D is an open-source library that supports rapid development of software that deals with 3D data. The Open3D frontend exposes a set of carefully selected data structures and algorithms in both C++ and Python. The backend is highly optimized and is set up for parallelization. In open 3d for converting our data into point cloud files, we use the following code "opcd.points = o3d.utility.Vector3dVector(test)". Here the parameter "test" is a numpy file which has coordinates of each point in x,y,z plane. To get coordinate data from images, we take X and Y as the coordinates of pixel coordinates which got segmented. The Z axis represents the value of the gap between slices of two images. Point cloud files are saved as filename.ply, ply if the extension of point cloud file. Once point cloud file is created we can view the same using open3D or matplotlib.

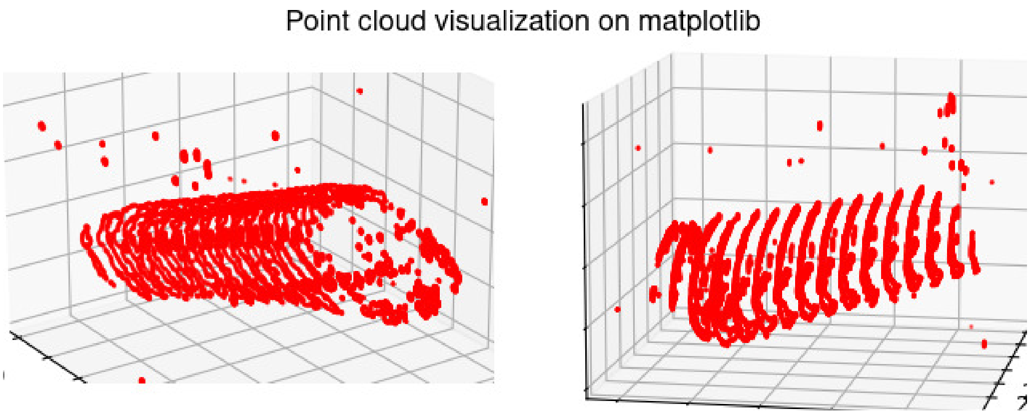

Figure 5.2: PCD Visualization in matplotlib

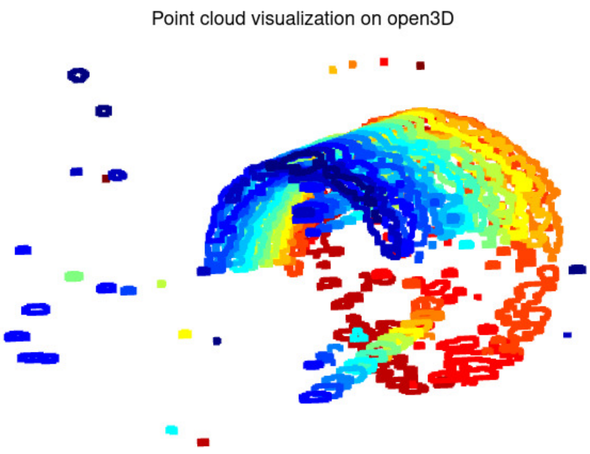

Figure 5.3: PCD Visualization in open3D

5.2.1 Dealing with image rotation

In some situations the image we capture from an ultrasound probe will have an angle component along with a distance component. Image will be rotated along the XYZ axis. So while plotting the image in 3D we should consider this factor too. For this we have to rotate each point cloud in a given angle. This can be done by iterating each points through the equation x'= x.cos()-ysin() and y'=x.sin()+ycos() The below image shows points rotated 60 degrees with respect to y axis

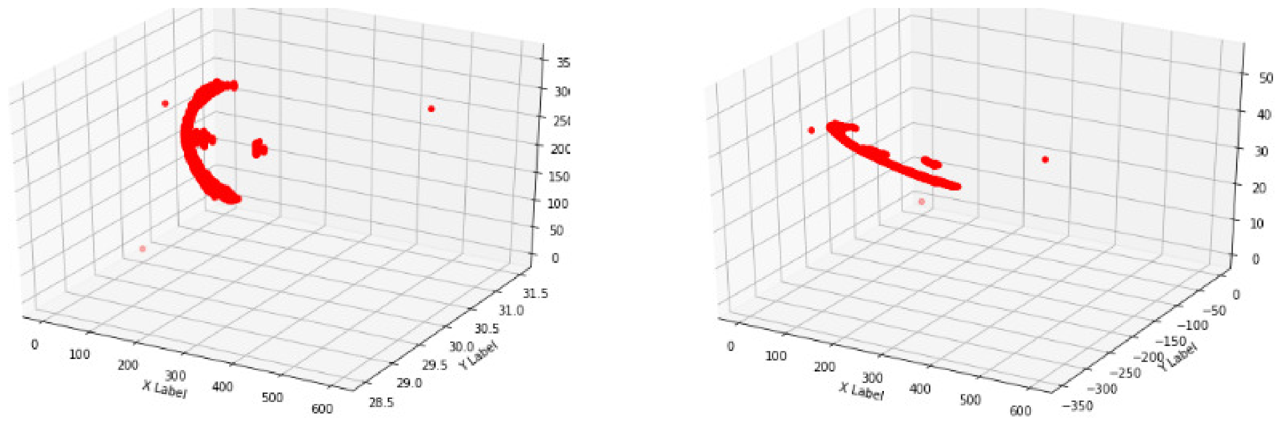

Figure 5.4: PCD Visualization in open3D

5.3 Conversion to 3D surface

While point clouds can be directly rendered and inspected,point clouds are often converted to polygon mesh or triangle mesh models, NURBS surface models, or CAD models through a process commonly referred to as surface reconstruction.There are many techniques for converting a point cloud to a 3D surface.Some approaches, like Delaunay triangulation, alpha shapes, and ball pivoting, build a network of triangles over the existing vertices of the point cloud, while other approaches convert the point cloud into a volumetric distance field and reconstruct the implicit surface so defined through a marching cubes algorithm. In our application we used a tool called "pyvista" this is a python library. It has in-built functions to deal with 3D processing. Here we use pyvista to interpolate the surfaces to create a smooth 3D appearance.

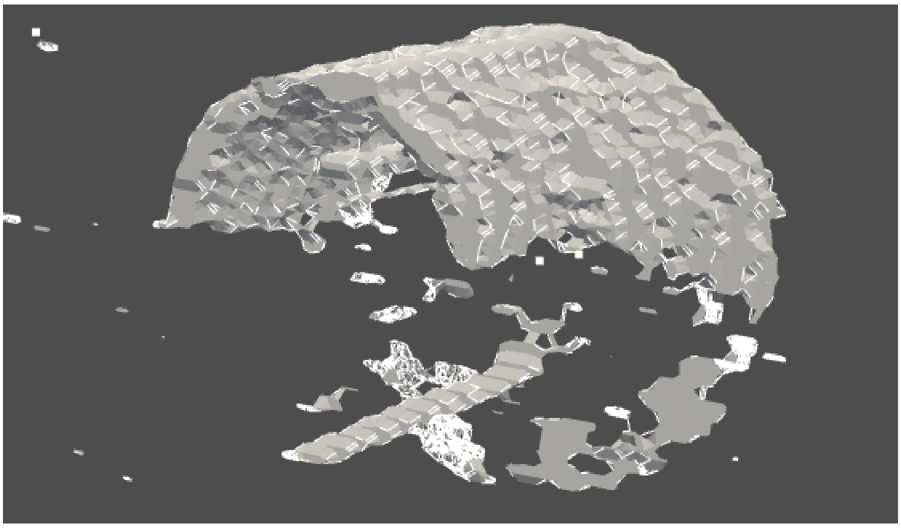

Figure 5.5: PCD Visualization in open3D

The picture above shows a 3D representation of the ultrasound sequence. The sequence was a picture of ultrasound slices. But lot of its data were missing when it was taken, that's why the 3D structure created looks incomplete

Chapter 6

Results

Through this miniproject we explored different methods of image processing techniques like binary thresholding, canny edge detection and K Means clustering to segment our region of interest and to create the 3D representation we have used two techniques.

In one technique we converted images to dicom format and used marching cubes algorithm in vtk library for 3D visualization.

In other technique the segmented images are converted into a point cloud file to get the 2D structure and the surface is reconstructed using library "pyvista"

Code used and results produced can be found at `https://drive.google.com/drive/folders/1FL9IAkpeXz8IOCLAraoVg8QnQW9Z1KT6?usp=sharing`

Chapter 7

Conclusion

Our goal was to generate 3D visualization from 2D ultrasound images. Through this work we were able gain insights on existing methods for 3D visualization of medical images as well as other open source techniques that can be used for the same . We were able to produce 3D visualization of provided ultrasound image sets using two approaches: VTK's marching cubes for isosurfaces and point cloud using pyvista .

Future Work

We can move forward to improve the results by applying machine learning techniques for generating intermediate slices, for obtaining smoother 3D volume and better visualization. Preprocessing of ultrasound images can also be focused on to produce better results.

Bibliography

[1] Nikolaus Knauer, Xose Luis Dean-Ben, and Daniel Razansky. "Spatial Compounding of Volumetric Data Enables Freehand Optoacoustic Angiography of Large-Scale Vascular Networks". In: *IEEE Transactions on Medical Imaging* PP (Oct. 2019), pp. 1–1. DOI: 10.1109/TMI.2019.2945297.

[2] Ralf Kohler. "A segmentation system based on thresholding". In: *Computer Graphics and Image Processing* 15.4 (1981), pp. 319–338.

[3] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[4] Mohammad Hamed Mozaffari and Won-Sook Lee. "Freehand 3-D Ultrasound Imaging: A Systematic Review". In: *Ultrasound in Medicine Biology* 43 (July 2017). DOI: 10.1016/j.ultrasmedbio.2017.06.009.

[5] Raphael Prevost et al. "3D freehand ultrasound without external tracking using deep learning". In: *Medical Image Analysis* 48 (2018), pp. 187–202. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2018.06.003. URL: https://www.sciencedirect.com/science/article/pii/S1361841518303712.

[6] Weibin Rong et al. "An improved CANNY edge detection algorithm". In: *2014 IEEE international conference on mechatronics and automation*. IEEE. 2014, pp. 577–582.

[7] VTK. *MedicalDemo1 vtk-examples/Python/Medical/MedicalDemo1*. URL: https://kitware.github.io/vtk-examples/site/Python/Medical/MedicalDemo1/.